

Programmation Web Serveur

Programmation orientée objet - Instances

D'après les cours de Philippe Renevier



Fabien Givors

Université de Nice Sophia Antipolis

Département Informatique

fabien.givors@unice.fr

CLASSE POUR LIER DONNÉES ET FONCTIONS

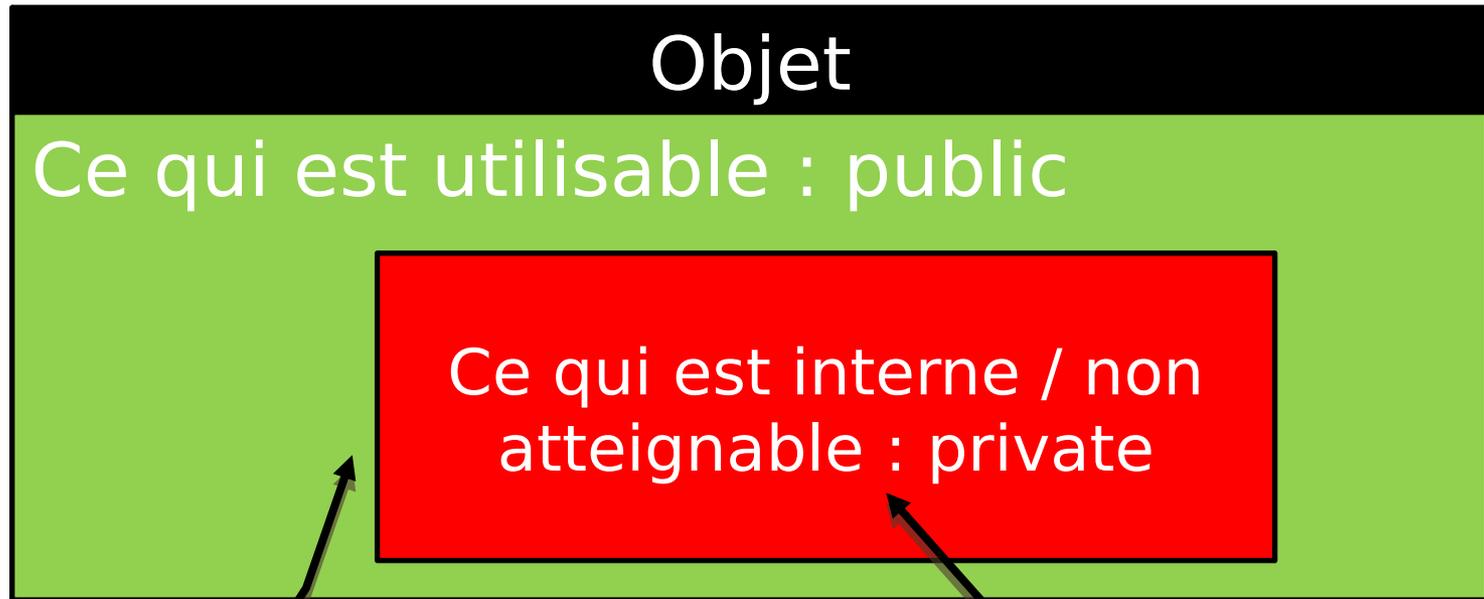
Données $\xleftrightarrow[\text{Valeur de retour}]{\text{Paramétrage,}}$ **Fonctions**

Objet
d'une
Classe

Champs de classe $\xleftrightarrow{\text{accès}}$ **Méthodes**

Les paramètres sont uniquement les valeurs externes à l'objet.
Moins de problème pour les valeurs de retour multiples

CLASSE POUR MASQUER



Défini comment l'objet est utilisé

Ce qui peut être changé sans modification de code pour l'utilisation de l'objet

ÉBAUCHE DE CLASSE MENU AVEC AFFICHAGE DES FORMULES

```
class Menu{
    // données initiales
    private $nom;
    // données supplémentaires
    private $formules = array();
    private $specialites = array();

    // constructeur
    public function Menu($nom)
    {
        $this->nom = $nom;
    }

    // Méthodes
    public function ajouteSpecialite($nom, $prix)
    {
        $this->specialite[] = array(
            "nom" => $nom,
            "prix" => int_val($prix)
        );
    }
    public function afficheFormules(){
        {
            ...
        }
    }
}

// Instanciation
$menu_ete = Menu("Carte d'été");
// Appel de méthodes
$menu_ete->ajouteSpecialite("Pan bagnat",4);
$menu_ete->afficheFormules();
```

Survol de XML, de l'API XML en php

UTILISATION DES OBJETS : CAS DU XML

EXEMPLE : HTML

- Hyper Text **Markup Language**
- Langage intermédiaire
- Interprété par le navigateur
- Ce qu'on voit à l'écran n'est pas le fichier
- HTML = structure sémantique
- Le navigateur sait sur quoi il peut « tomber », il sait comment l'interpréter

HTML = STRUCTURE SÉMANTIQUE DU DOC

Le dormeur du val. C'est un trou de verdure où chante une rivière Accrochant follement aux herbes des haillons D'argent ; où le soleil, de la montagne fière, Luit : c'est un petit val qui mousse de rayons. Un soldat jeune, bouche ouverte, tête nue, Et la nuque baignant dans le frais cresson bleu, Dort ; il est étendu dans l'herbe, sous la nue, Pâle dans son lit vert où la lumière pleut. Les pieds dans les glaïeuls, il dort. Souriant comme Sourirait un enfant malade, il fait un somme : Nature, berce-le chaudement : il a froid. Les parfums ne font pas frissonner sa narine ; Il dort dans le soleil, la main sur sa poitrine, Tranquille. Il a deux trous rouges au côté droit.

Le dormeur du val

C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent ; où le soleil, de la montagne fière,
Luit : c'est un petit val qui mousse de rayons.

Un soldat jeune, bouche ouverte, tête nue,
Et la nuque baignant dans le frais cresson bleu,
Dort ; il est étendu dans l'herbe, sous la nue,
Pâle dans son lit vert où la lumière pleut.

Les pieds dans les glaïeuls, il dort. Souriant comme
Sourirait un enfant malade, il fait un somme :
Nature, berce-le chaudement : il a froid.

Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil, la main sur sa poitrine,
Tranquille. Il a deux trous rouges au côté droit.

<h1>Le dormeur du val**</h1>**

<p>C'est un trou de verdure où chante une
rivière**
**

Accrochant follement aux herbes des haillons**
**
D'argent ; où le soleil, de la montagne fière, **
**
Luit : c'est un petit val qui mousse de rayons.**</p>**

<p>Un soldat jeune, bouche ouverte, tête nue, **
**
Et la nuque baignant dans le frais cresson bleu, **
**
Dort ; il est étendu dans l'herbe, sous la nue, **
**
Pâle dans son lit vert où la lumière pleut.**</p>**

<p>Les pieds dans les glaïeuls, il dort. Souriant
comme**
**
Sourirait un enfant malade, il fait un somme : **
**
Nature, berce-le chaudement : il a froid.**</p>**

<p>Les parfums ne font pas frissonner sa narine ; **
**
/>Il dort dans le soleil,
la main sur sa poitrine, **
**
Tranquille. Il a deux trous rouges au côté droit.**</p>**

POURQUOI UN LANGAGE À BALISE ?

- Texte brut :

 - Nice 15 22

- Texte avec balises :

 - `<meteo>`

 - `<ville>Nice</ville>`

 - `<matin>15</matin>`

 - `<midi>22</midi>`

 - `</meteo>`

- XML : méta-modèle (modèle pour langage à Balise)

LE XML ?

- Déterminer un langage (DTD)
- Instancier ce langage
 - Document web = autant d'instance d'une DTD web
 - (pas forcément XML, mais néanmoins très proche)
- Voir les documents comme des arbres d'objet :
DOCUMENT OBJECT MODEL

STRUCTURE SÉMANTIQUE HTML EN ARBRE

Le dormeur du val

C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent ; où le soleil, de la montagne fière,
Luit : c'est un petit val qui mousse de rayons.

Un soldat jeune, bouche ouverte, tête nue,
Et la nuque baignant dans le frais cresson bleu,
Dort ; il est étendu dans l'herbe, sous la nue,
Pâle dans son lit vert où la lumière pleut.

Les pieds dans les glaïeuls, il dort. Souriant comme
Sourirait un enfant malade, il fait un somme :
Nature, berce-le chaudement : il a froid.

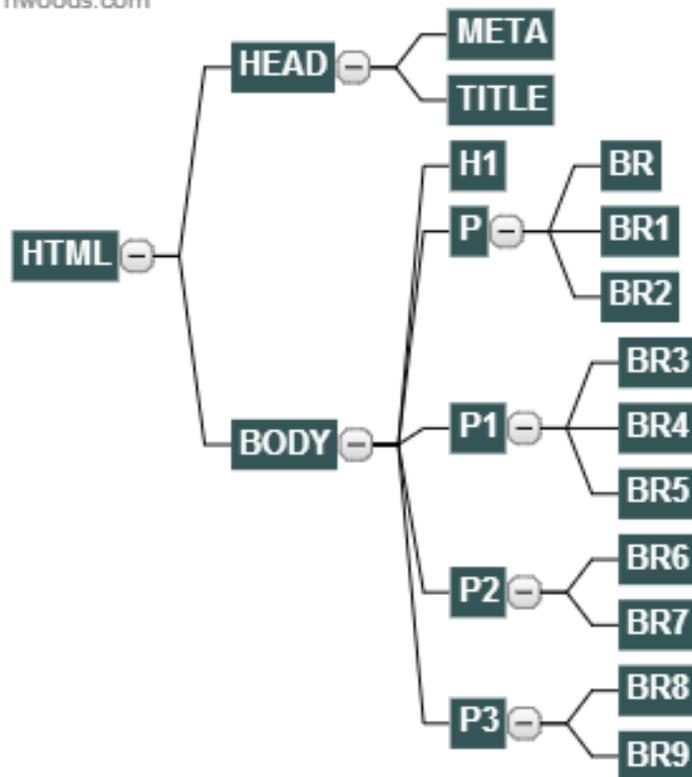
Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil, la main sur sa poitrine,
Tranquille. Il a deux trous rouges au côté droit.

GoJS evaluation

(c) 1998-2014 Northwoods Software

Not for distribution or production use

nwoods.com



Le dormeur du val

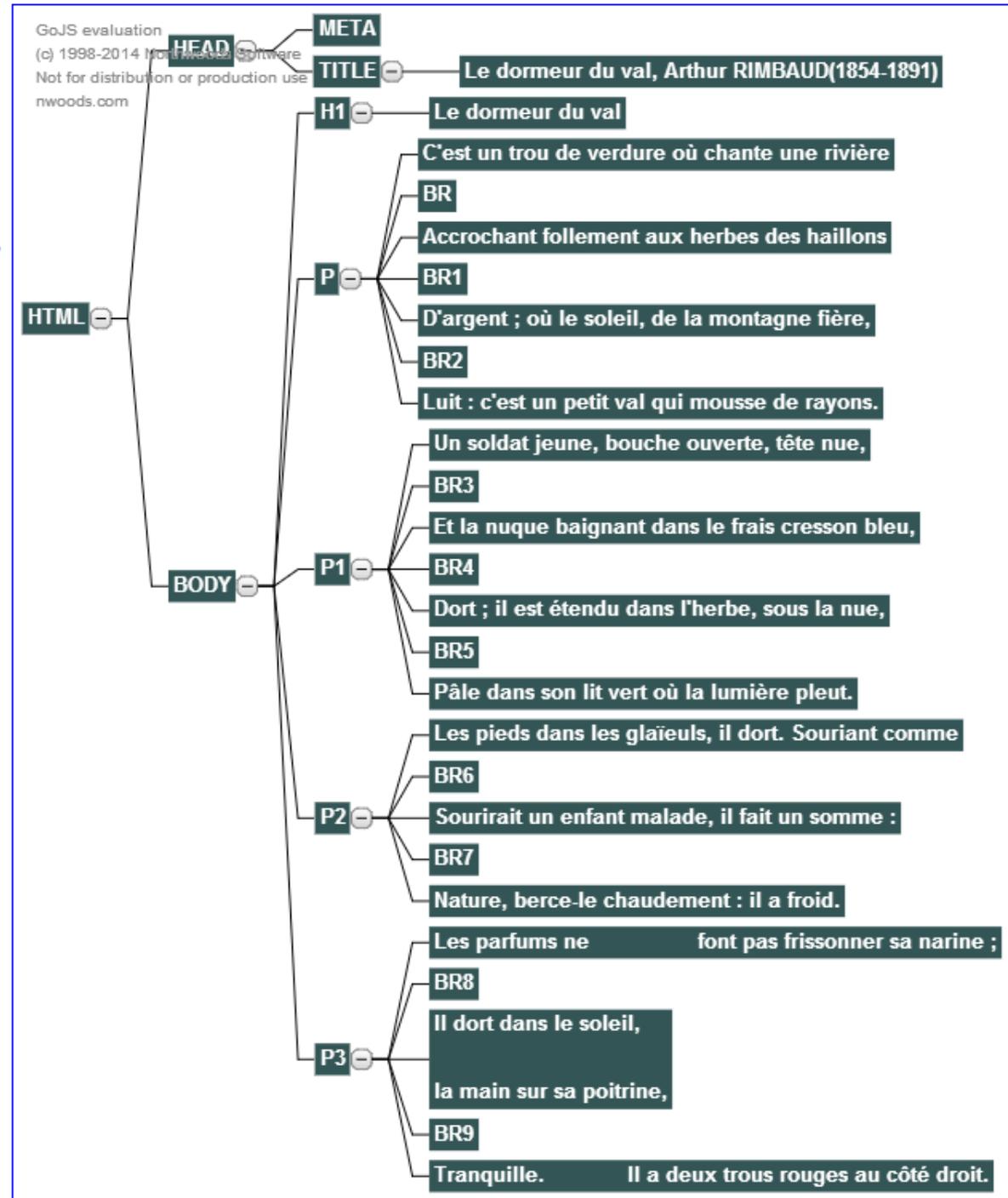
C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent ; où le soleil, de la montagne fière,
Luit : c'est un petit val qui mousse de rayons.

Un soldat jeune, bouche ouverte, tête nue,
Et la nuque baignant dans le frais cresson bleu,
Dort ; il est étendu dans l'herbe, sous la nue,
Pâle dans son lit vert où la lumière pleut.

Les pieds dans les glaïeuls, il dort. Souriant
comme
Sourirait un enfant malade, il fait un somme :
Nature, berce-le chaudement : il a froid.

Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil, la main sur sa poitrine,
Tranquille. Il a deux trous rouges au côté droit.

ARBRE
AVEC
LES
NŒUDS
"TEXT"
UTILES



LECTURE D'UN XML PAR PHP

- API DOMDocument
- Un Document :
 - DOMDocument
 - C'est aussi un nœud (DOMNode) => parcours selon un nœud possible
 - De nombreuses propriétés et de nombreuses méthodes
- création

```
// Create a new DOM Document to hold our document structure
$xml = new DOMDocument();
$xml->load ("fichier.xml");
```

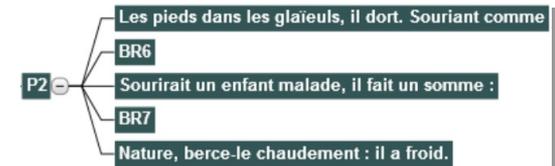
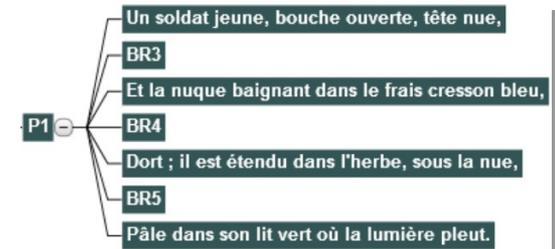
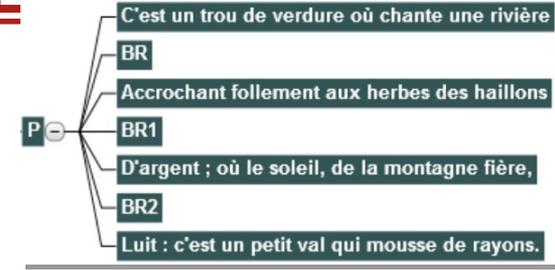
- mixed load (string \$filename_OU_Url [, int \$options = 0])
- bool loadHTML (string \$source)
- bool loadHTMLFile (string \$filename)
- mixed loadXML (string \$source [, int \$options = 0])

SÉLECTION DE BALISES DU DOMDOCUMENT

- ▣ DOMElement getElementById (string \$elementId)
- ▣ DOMNodeList getElementsByTagName (string \$name)

EXEMPLE DE GETELEMENTSBYTAGNAME

```
1 <?php
2 $doc = new DOMDocument();
3 $doc->loadHTMLFile("poesie.html");
4 $listeP = $doc->getElementsByTagName("p");
5 ?>
6 <!doctype html>
7 <html>
8 <head>
9     <meta charset="utf-8" />
10    <title>exemple dom et php</title>
11 </head>
12 <body>
13 <?php
14 foreach($listeP as $p)
15 {
16     echo "<hr />{$p->textContent} <hr />";
17 }
18 ?>
19 </body>
20 </html>
21
```



UTILISATION D'UN DOMELEMENT (1/3)

- C'est aussi un DOMNode
 - Parcours selon un nœud
- Création : plutôt par un objet DOMDocument, sinon avec un nom de tag
- Méthodes liées aux attributs
 - string getAttribute (string \$name)
 - bool hasAttribute (string \$name)
- sélection
 - DOMNodeList getElementsByTagName (string \$name)

UTILISATION D'UN DOMÉLEMENT (2/3)

▫ Élément de base de l'arbre, avec parent, fratrie, enfants, etc.

// nom du tag

public readonly string *\$nodeName* ;

// valeur (attention, pas toujours pertinent, dépend du type de nœud)

public string *\$nodeValue* ;

// type : un entier prédéfini :

<http://php.net/manual/fr/dom.constants.php>

// ex: 3 == XML_TEXT_NODE == du texte...

public readonly int *\$nodeType* ;

// Cet attribut retourne le contenu texte de ce nœud et de ses descendants.

public string *\$textContent* ;

UTILISATION D'UN DOMELEMENT (3/3)

▫ Attributs de structure, pour la navigation

```
// nœud parent
```

```
public readonly DOMNode $parentNode ;
```

```
// listes des nœuds enfants
```

```
public readonly DOMNodeList $childNodes ;
```

```
// premier nœud inclus
```

```
public readonly DOMNode $firstChild ;
```

```
// dernier nœud inclus
```

```
public readonly DOMNode $lastChild ;
```

```
// nœud « frère » (ayant le même parent) précédent
```

```
public readonly DOMNode $previousSibling ;
```

```
// nœud « frère » (ayant le même parent) suivant
```

```
public readonly DOMNode $nextSibling ;
```

STRUCTURATION DOM: DOMNODELIST

- Juste une liste de nœuds

```
DOMNodeList {
```

```
/* Propriétés */
```

```
readonly public int $length ;
```

```
/* Méthodes */
```

```
DOMNode DOMNodelist::item ( int $index )
```

```
}
```

- exemples

- for(\$i = 0; \$i < \$list->length; \$i++) { ... }

- if (\$list->item(0)->nodeName == "item") { ... }

FLUX RSS : DÉFINITION ET EXEMPLE

LE FLUX RSS, DES EXEMPLES

- Des exemples :
 - <http://www.lemonde.fr/rss/>
 - <http://www.leparisien.fr/services/rss/>
 - <http://www.liberation.fr/rss/>
 - http://filuns.unice.fr/accueil/@@rss_view
- Un format d'échange d'information

DEFINITION D'UN FLUX RSS (1/2)

- Un flux `<rss>` est composé d'au moins un `<channel>` (généralement 1)
- Un `<channel>` est composé
 - `<title>` : Définit le titre du flux ;
 - `<description>` : Décrit succinctement le flux ;
 - `<link>` : Définit l'URL du site correspondant au flux.
 - Etc. (pour les informations)
 - Et d'un ou plusieurs `<item>`

DEFINITION D'UN FLUX RSS (2/2)

- Un `<item>` est composé
 - `<title>` : Définit le titre de l'actualité ;
 - `<link>` : Définit l'URL du flux correspondant à l'actualité ;
 - `<pubDate>` : Définit la date de l'actualité ;
 - `<description>` : Définit une description succincte de l'actualité ;
 - `<guid>` : Définit de manière unique l'actualité.
- Voire aussi, pour `<item>`
 - `<author>` : Définit l'adresse électronique (mail) de l'auteur ;
 - `<category>` : Associe l'item à une catégorie ;
 - Etc.

EXPLOITER UN FLUX : BESOIN D'UNE API

- Convertir le flux rss (du xml = du texte) en objet
- API « DOM » en php
 - Abstract programming interface
 - Une définition « publique » de comment utiliser des éléments de programmation (comme nous le faisons)
 - <http://www.php.net/manual/fr/book.dom.php>

LIRE UN FLUX... (À DISTANCE)

```
// utilisation d'un objet de type DOMDocument
$this->xml = new DOMDocument();

// le @ est un suppress warning
// le paramètre de load est un nom de fichier (chemin) ou
une url
@$this->xml->load($this->url);

// exemple : obtenir tous les articles (= <item>) du flux
$items = $this->xml->getElementsByTagName("item");
// $items est la liste de tous les <item> dans le flux
ouvert
```

TRANSFORMER DU XML EN HTML (1/3)

- Principe :
 - extraire les informations
 - navigation (getElementsByTagName, ...)
 - obtenir la valeur (textContent, ...)
 - les utiliser pour remplir un modèle
 - remplir autant de modèle que nécessaire
- Réalisation :
 - savoir l'HTML à produire
 - itérer sur les informations.

TRANSFORMER DU XML EN HTML

(2/3)

▫ Html ciblé (c.f. tp)

```
<div class='article'>
  <h2>titre</h2>
  <p>auteur</p>
  description...
</div>
```

▫ Côté php

```
$rssHtml = "<div
class='article'><h2>$titre<
/h2><p>$auteur</p>";

$rssHtml .=
"$description<br /><strong
class='date'>".date("Y-m-d
[h:i]", strtotime($dt-
>textContent))."</strong>\n
</div>";

// ici description +
date...
```

TRANSFORMER DU XML EN HTML

(3/3)

```
@$items = $this->xml->getElementsByTagName("item");
if ($items && ($items->length > 0)) {
    // foreach($items as $item)
    for($i = 0; $i < $nb; $i++) {
        $item = $items->item($i);
        $dt = $item->getElementsByTagName("pubDate")->item(0);

        $titre = $item->getElementsByTagName("title")->item(0)->textContent;

        if ( $item->getElementsByTagName("author")->length > 0 ) {
            $auteur = $item->getElementsByTagName("author")->item(0);
        }
        else $auteur = $auteur_par_defaut;

        $description = $item->getElementsByTagName("description")->item(0);
        $description = $description->textContent;

        $rssHtml .= "<div class='article'><h2>$titre</h2><p>$auteur</p>";
        $rssHtml .= "$description<br /><strong class='date'>".date("Y-m-d [h:i]", strtotime($dt-
>textContent))."</strong>\n</div>";
        // ici description + date...
    }
}
```

LIRE UN FLUX... AVEC UN CACHE

- ▣ Enregistrer les données dans un fichier
- ▣ Lire depuis ce fichier
- ▣ Pour mettre à jour le fichier, comparer sa date de dernière modification avec
 - ▣ Soit un délai avant mise à jour
 - ▣ Soit avec une date de mise à jour sur le serveur (plus compliqué)

IMPORTER LES ARTICLES DANS VOTRE SITE

- Idée : lire une flux rss et l'incorporer dans le site
 - Attention à la légalité
- Le flux rss est-il complet ?
 - Faut-il préciser un auteur ?
 - Faut-il préciser une catégorie ?
 - Etc.
- Mise à jour ? Les doublons ? Etc.